**Appears in the** *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining, Nov. 2002, Baltimore, MD.*

# Relational Data Mining with Inductive Logic Programming for Link Discovery

Raymond J. Mooney, Prem Melville, Lappoon Rupert Tang
Department of Computer Sciences
University of Texas
Austin, TX 78712-1188
{mooney,melville,rupert}@cs.utexas.edu

Jude Shavlik, Inês de Castro Dutra, David Page, Vítor Santos Costa
Department of Biostatistics and Medical Informatics and
Department of Computer Sciences
University of Wisconsin
Madison, WI 53706-1685
{shavlik,dpage}@cs.wisc.edu, {dutra,vitor}@biostat.wisc.edu

## Abstract

*Link discovery (LD) is an important task in data mining for counter-terrorism and is the focus of DARPA's Evidence Extraction and Link Discovery (EELD) research program. Link discovery concerns the identification of complex relational patterns that indicate potentially threatening activities in large amounts of relational data. Most data-mining methods assume data is in the form of a feature-vector (a single relational table) and cannot handle multi-relational data. Inductive logic programming is a form of relational data mining that discovers rules in first-order logic from multi-relational data. This paper discusses the application of ILP to learning patterns for link discovery.*

## 1 Introduction

Since the events of September 11, 2001, the development of information technology that could aid intelligence agencies in their efforts to detect and prevent terrorism has become an important focus of attention. The Evidence Extraction and Link Discovery (EELD) program of the Defense Advanced Research Projects Agency (DARPA) is one research project that attempts to address this issue. The establishment of the EELD program for developing advanced software for aiding the detection of terrorist activity predates the events of 9/11. The program had its genesis at a preliminary DARPA planning meeting held at Carnegie Mellon University after the opening of the Center for Automated Learning and Discovery in June of 1998. This meeting discussed the possible formation of a new DARPA research program focused on novel knowledge-discovery and data-mining (KDD) methods appropriate for counter-terrorism.

The scope of the new program was subsequently expanded to focus on three related sub-tasks in detecting potential terrorist activity from numerous large information sources in multiple formats. *Evidence extraction* (EE) is the task of obtaining structured evidence data from unstructured, natural-language documents. EE builds on information extraction technology developed under DARPA's earlier MUC (Message Understanding Conference) programs [23, 8] and the current ACE (Automated Content Extraction) program at the National Institute of Standards and Technology (NIST)[29]. *Link Discovery* (LD) is the task of identifying known, complex, multi-relational patterns that indicate potentially threatening activities in large amounts of relational data. Some of the input data for LD comes from EE, other input data comes from existing relational databases. Finally, *Pattern Learning* (PL) concerns the automated discovery of new relational patterns for potentially threatening activities. Novel patterns learned by PL can be used to improve the accuracy of LD. The current EELD pro-

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **NOV 2002** | | **00-00-2002 to 00-00-2002** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **Relational Data Mining with Inductive Logic Programming for Link Discovery** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Texas at Austin,Department of Computer Sciences,Austin,TX,78712** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES
**U.S. Government or Federal Rights License**

14. ABSTRACT
**Link discovery (LD) is an important task in data mining for counter-terrorism and is the focus of DARPA's Evidence Extraction and Link Discovery (EELD) research program. Link discovery concerns the identi cation of complex relational patterns that indicate potentially threatening activities in large amounts of relational data. Most data-mining methods assume data is in the form of a feature-vector (a single relational table) and cannot handle multi-relational data. Inductive logic programming is a form of relational data mining that discovers rules in rst-order logic from multi-relational data. This paper discusses the application of ILP to learning patterns for link discovery.**

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | |
| **unclassified** | **unclassified** | **unclassified** | **Same as Report (SAR)** | **8** | |

gram focused on these three sub-topics started in the summer of 2001. After 9/11, it was incorporated under the new Information Awareness Office (IAO) at DARPA.

The data and patterns used in EELD include representations of people, organizations, objects, and actions and many types of relations between them. The data is perhaps best represented as a large graph of entities connected by a variety of relations. The areas of *link analysis* and *social network analysis* in sociology, criminology, and intelligence [19, 37, 33] study such networks using graph-theoretic representations. Data mining and pattern learning for counter terrorism therefore requires handling such multi-relational, graph-theoretic data.

Unfortunately, most current data-mining methods assume the data is from a single relational table and consists of flat tuples of items, as in market-basket analysis. This type of data is easily handled by machine learning techniques that assume a "propositional" (a.k.a "feature vector" or "attribute value") representation of examples [41]. *Relational data mining* (RDM) [14], on the other hand, concerns mining data from multiple relational tables that are richly connected. Given the style of data needed for link discovery, pattern learning for link discovery requires *relational* data mining. The most widely studied methods for inducing relational patterns are those in *inductive logic programming* (ILP) [27, 22]. ILP concerns the induction of Horn-clause rules in first-order logic (i.e., logic programs) from data in first-order logic. This paper discusses our on-going work on applying ILP to link discovery as a part of the EELD project.

## 2 Inductive Logic Programming (ILP)

ILP is the study of learning methods for data and rules that are represented in first-order predicate logic. Predicate logic allows for quantified variables and relations and can represent concepts that are not expressible using examples described as feature vectors. A relational database can be easily translated into first-order logic and be used as a source of data for ILP [44]. As an example, consider the following rules, written in Prolog syntax (where the conclusion appears first), that define the uncle relation:

```
uncle(X,Y) :- brother(X,Z),parent(Z,Y).
uncle(X,Y) :- husband(X,Z),sister(Z,W),
              parent(W,Y).
```

The goal of *inductive logic programming* (ILP) is to infer rules of this sort given a database of background facts and logical definitions of other relations [27, 22]. For example, an ILP system can learn the above rules for uncle (the *target predicate*) given a set of positive and negative examples of uncle relationships and a set of facts for the relations parent, brother, sister, and husband (the *background predicates*) for the members of a given extended family, such as:

```
uncle(tom,frank), uncle(bob,john),
¬uncle(tom,cindy), ¬uncle(bob,tom)
parent(bob,frank), parent(cindy,frank),
parent(alice,john), parent(tom,john),
brother(tom,cindy), sister(cindy,tom),
husband(tom,alice), husband(bob,cindy).
```

Alternatively, rules that logically define the brother and sister relations could be supplied and these relationships inferred from a more complete set of facts about only the "basic" predicates: `parent`, `spouse`, and `gender`.

If-then rules in first-order logic are formally referred to as *Horn clauses*. A more formal definition of the ILP problem follows:

- **Given:**
  - Background knowledge, $B$, a set of Horn clauses.
  - Positive examples, $P$, a set of Horn clauses (typically ground literals).
  - Negative examples, $N$, a set of Horn clauses (typically ground literals).
- **Find:** A hypothesis, $H$, a set of Horn clauses such that:
  - $\forall p \in P : H \cup B \models p$ (completeness)
  - $\forall n \in N : H \cup B \not\models n$ (consistency)

A variety of algorithms for the ILP problem have been developed [13] and applied to a variety of important data-mining problems [12]. Nevertheless, relational data mining remains an under-appreciated topic in the larger KDD community. For example, recent textbooks on data mining [17, 41, 18] hardly mention the topic. Therefore, we believe it is an important topic for "next generation" data mining systems. In particular, it is critical for link discovery applications in counter-terrorism.

## 3 Initial Work on ILP for Link Discovery

We tested different ILP algorithms on various EELD datasets. The current EELD datasets pertain to two domains — Nuclear Smuggling and Contract Killing. The Contract-Killing domain is further divided into natural (real world) data manually collected and extracted from news sources and synthetic (artificial) data generated by a simulator. Section 3.1 presents our experimental results on the natural Smuggling and Contract-Killing data, while section 3.2 presents our initial results on the synthetic Contract-Killing data.

## 3.1 Experiments on Natural Data

### 3.1.1 The Nuclear-Smuggling Data

The Nuclear-Smuggling dataset consists of reports on Russian nuclear materials smuggling [24]. The Chronology of Nuclear and Radioactive Smuggling Incidents is the basis for the analysis of patterns in the smuggling of Russian nuclear materials. The information in the Chronology is based on open-source reporting, primarily World News Connection (WNC) and Lexis-Nexis. There are also some articles obtained from various sources that have been translated from Italian, German and Russian. The research from which the Chronology grew began in 1994 and the chronology itself first appeared as an appendix to a paper by Williams and Woessner in 1995 [40, 39]. The continually evolving Chronology then was published twice as separate papers in the same journal as part of the "Recent Events" section [42, 43]. As part of the Evidence Extraction and Link Discovery (EELD) project, the coverage of the Chronology was extended to March 2000 and the Chronology itself grew to 572 incidents. The incident descriptions in the Chronology are one entry descriptions per incident. The incidents in the Chronology have also been extensively cross-referenced.

The data is presented as a chronology of the incidents in a relational database format. This format contains Objects (described in rows in tables), each of which has Attributes of differing types (i.e., columns in the tables), the values of which are a matter of input from the source information or from the user. The Objects are of different types, which are denoted by prefixes (E_, EV_, LK_, and L_), and consist of the following.

- Entity Objects (E_...): these consist of E_LOCATION, E_MATERIAL, E_ORGANIZATION, E_PERSON, E_SOURCE, and E_WEAPON;
- Event Objects (EV_...): these currently consist of the generic EV_EVENT;
- Link Objects (LK_...): used for expressing links between/among Entities and Events, and currently consisting of those represented by X's in Table 3.1.1.

The actual database we use in our experiments has over 40 relational tables. The number of tuples in a relational table vary from 800 to as little as 2 or 3 elements.

The ILP system has to learn which events in an incident are *related* in order to construct larger knowledge structures that can be recognized as threats. Hence the ILP system needs positive training examples that specify "links" between events. We assume all other events are unrelated and therefore compose a set of negative examples. We stipulate that *related* is commutative. Therefore we specified to the ILP system used in our experiments that `related(B,A)` is true if `related(A,B)` is proven, and vice-versa. Our set of examples consists of 143 positive examples and 517 negative examples.

The linking problem in the Nuclear-Smuggling data is thus quite challenging in that it is a heavily relational learning problem over a large number of relations, whereas traditional ILP applications usually require a small number of relations.

### 3.1.2 The Natural Contract-Killing Data

The dataset of contract killings was first compiled by O'Hayon and Cook [6]. It was a response to research on Russian organized crime that encountered frequent and often tantalizing references to contract killings. Each of the contract-killing reports provided a still photograph of the criminal scene in Russia, but there was no comparable assessment of how these were linked, what the trends were, who the victims were, the relationship between victims themselves or the relationship between victims and perpetrators. The dataset on contract killings has been continually expanded by Cook and O'Hayon with funding from DARPA's EELD program through Veridian Systems Division (VSD) [38]. The database was captured as a "chronology" of the incidents. Each incident in the chronology received a description of the information drawn from the sources, typically one news article, but occasionally more than one. As in the Nuclear-Smuggling dataset, information in the chronology is based on open-source reporting, especially Foreign Broadcast Information Service (FBIS) and Joint Publications Research Service (JPRS) journals, and subsequently both FBIS on-line and the cut-down on-line version World News Connection (WNC). These services and Lexis-Nexis are the main information sources. Additional materials on the worldwide web were consulted when this was feasible and helpful. The search was as exhaustive as possible given the limited time and resources of those involved.

The data is organized in relational tables in the same format as the Nuclear-Smuggling data described in the previous section. The dataset used in our experiments has 48 relational tables. The number of tuples in a relational table varies from 1,000 to as little as 1 element. The ILP learner task was to characterize Rival versus Obstacle plus Threat events (i. e., the Obstacle and Threat examples were pooled into one category, thereby producing a two-category learning task). Rival, Obstacle, and Threat are treated as "motives" in the dataset. The motivation to this learning task thus is to recognize patterns of activity that indicate underlying motives, which in turn contributes to recognizing threats. The number of positive examples in this dataset is 38, while the number of negative examples is 34.

**Table 1. Links among Entities and Events in Nuclear-Smuggling Data**

| | Event | Person | Organization | Location | Weapon | Material |
|---|---|---|---|---|---|---|
| Event | X | | | | | |
| Person | X | X | | | | |
| Organization | X | X | X | | | |
| Location | X | X | X | X | | |
| Weapon | X | X | X | X | X | |
| Material | X | X | X | X | X | X |

#### 3.1.3 ILP Results

**Aleph** We use the ILP system Aleph [35] in some of our experiments, those involving natural, rather than synthetic, data. By default, Aleph uses a simple greedy set covering procedure that constructs a complete and consistent hypothesis one clause at a time. In the search for any single clause, Aleph selects the first uncovered positive example as the seed example, "saturates" this example, and performs an admissible search over the space of clauses that subsume this saturation, subject to a user-specified clause length bound. Further details about our use of Aleph in these experiments are available in [11].

**Ensembles** *Ensembles* aim at improving accuracy through combining the predictions of multiple classifiers in order to obtain a single classifier. Therefore, we also investigate employing an ensemble of classifiers, where each classifier is a logical theory generated by Aleph. Many methods have been presented for ensemble generation [10]. In this paper, we concentrate on a popular method that is known to frequently create a more accurate ensemble than individual components, *bagging* [1]. Bagging works by training each classifier on a random sample from the training set. Bagging has the important advantage that it is effective on "unstable learning algorithms" [2], where small variations in parameters can cause huge variations in the learned theories. This is the case with ILP. A second advantage is that it can be implemented in parallel trivially. Further details about our bagging approach within ILP, as well as our experimental methodology, can be found in [11]. Our experimental results are based on a five-fold cross-validation, where five times we train on 80% of the examples and then test what was learned on the remaining 20% (in addition, each example is in one and only one test set).

For the task of identifying linked events within the Nuclear-Smuggling dataset, Aleph produces an average testset accuracy of 83%. This is an improvement over the baseline case (majority class—always guessing two events are not linked), which produces an average accuracy of 78%. Bagging (with 25 different sets of rules) increases the accuracy to 86%.

An example of a rule with good accuracy found by the system is shown in Figure 1. This rule covers 39 of the 143 positive examples and no negative examples.

According to this rule, two smuggling events `A` and `D` are related if they involve two people `C` and `E` and these two people are connected to a third person through a third event `F` that has the same person-person motive description, and the same dates. The "_" symbols mean that those arguments were not relevant for that rule.

The task of identifying motive in the Contract-Killing data set is much more difficult, with Aleph's accuracy at 56%, compared with the baseline accuracy of 50%. Again bagging improves the accuracy, this time to 63%. The rule in Figure 2 shows one kind of logical clause the ILP system we use found for this dataset.

The rule covers 19 of the 38 positive examples and a single negative example. The rule says that event `A` is a killing by a rival if we can follow a chain of events that connects event `A` to event `B`, event `B` to event `E`, and event `E` to an event `F` that relates two organizations. Events `A` and `E` have the same kind of relation, `RelationC`, to `B`. All events in the chain are subsets of the same incident `D`.

### 3.2 Experiments on Synthetic Data

#### 3.2.1 The Synthetic Contract-Killing Data

The synthetic data for Contract Killing was generated by a Bayesian Network (BN) simulator based on a probabilistic model developed by Information Extraction and Transport Incorporated (IET). The BN simulator outputs case files, which contain complete and unadulterated descriptions of each murder case. These case files are then filtered for observability, so that facts that would not be accessible to an investigator are eliminated. To make the task more realistic this data is also corrupted, e.g., by misidentifying role players or incorrectly reporting group memberships. This filtered and corrupted data form the evidence files. In the evidence files, facts about each event are represented as predicates, such as:

```
linked(EventA,EventD) :-
     lk_event_person(_,EventA,PersonC,_,RelationB,RelationB,_),
     lk_person_person(_,PersonC,_,EventF,_,_,_,MotiveG,StartDateH,EndDateI,DateDescriptionJ),
     lk_event_person(_,EventD,PersonE,_,RelationB,RelationB,_),
     lk_person_person(_,PersonE,_,EventF,_,_,_,MotiveG,StartDateH,EndDateI,DateDescriptionJ).
```

**Figure 1. Nuclear-Smuggling Data: Sample Learned Rule**

```
rivalKilling(EventA) :-
     lk_event_event(_,EventB,EventA,RelationC,EventDescriptionD),
     lk_event_event(_,EventB,EventE,RelationC,EventDescriptionD),
     lk_event_event(_,EventE,EventF,_,EventDescriptionD),
     lk_org_org(_,_,_,EventF,_,_,_,_,_).
```

**Figure 2. Natural Contract-Killing Data: Sample Learned Rule**

```
isa(Murder714, MurderForHire)
perpetrator(Murder714, Killer186)
victim(Murder714, MurderVictim996)
deviceTypeUsed(Murder714, PistolCzech)
```

The synthetic contract killing dataset that we used consists of 200 murder events. Each murder event has been labeled as a murder for hire, first-degree or second-degree murder. There are 71 murder for hire events, 75 first-degree and 54 second-degree murder events. Our task was to learn a classifier to correctly classify an unlabeled event into one of these three categories.

#### 3.2.2 ILP Results

For this task, we used a variation of mFoil [22] to learn a binary classifier to discriminate between events that are murder for hire and events that are not. Like Aleph, mFoil learns one clause at a time using greedy covering, but uses a constrained, general-to-specific search to learn individual rules. We also used mFoil to learn two more classifiers to identify first-degree and second-degree murders. The three binary classifiers are combined to form a three-way classifier for the task. If an event is classified as a positive example by only one classifier then the event is labeled with the category corresponding to that classifier. If more than one classifier classifies an event as a positive example then we select the category more commonly represented in the training data.

We ran 10-fold cross-validation on the dataset of 200 murder events. We measured the precision and recall of our classifier for each of the three categories. Precision and recall for a category is defined below:

$$Precision_C = \frac{number\ of\ events\ correctly\ classified\ as\ C}{number\ of\ events\ classified\ as\ C} \times 100\%$$

$$Recall_C = \frac{number\ of\ events\ correctly\ classified\ as\ C}{number\ of\ C\ events} \times 100\%$$

The results are summarized in Table 2. We observe that apart from recall for second-degree murders, the precision and recall results are all above 85%. Our system learns a very precise classifier for second-degree murders, but as a consequence it has a lower recall. However, we can adjust the parameters of our system to compromise precision for higher recall.

We also computed the accuracy of our classifier, which is defined as the percentage of events correctly classified into one of the three categories. We compare this to the majority-class classifier, which always classifies events as the most frequently represented category. In our experiments the accuracy of the majority-class classifier is 38%. And the classification accuracy of our system is 77% which is more than twice that of the majority-class classifier.

Figure 3 shows some of the sample rules that our system learns. According to the first rule a murder event that involves a member of a criminal organization and that is associated with another crime that was motivated by economic gains is a murder for hire. The second rule says that if a murder is the result of an event that was performed by someone in love, then it is a first-degree murder (as these are mainly premeditated murders). According to the third rule if a murder is the result of a theft that is motivated by rivalry and that is performed on public property then it is a second-degree murder. These sample rules show that not only does our system do well in classifying the different events, it also produces rules that are meaningful and interpretable by humans.

**Table 2. Results on the Synthetic Contract-Killing Data**

| | **Murder for hire** | **1st degree** | **2nd degree** |
|---|---|---|---|
| **Precision** | 86% | 91% | 96% |
| **Recall** | 91% | 88% | 59% |

```
murderForHire(A) :-
  groupMemberMaleficiary(A, B),
  subEvents(A, C),
  crimeMotive(C, economic).

firstDegreeMurder(A) :-
  subEvents(A, B),
  performedBy(B, C),
  loves(C, D).

secondDegreeMurder(A) :-
  subEvents(A, B),
  eventOccursAtLocationType(B, publicProperty),
  crimeMotive(B, rival),
  occurrentSubeventType(B, stealing_Generic).
```

**Figure 3. Synthetic Contract-Killing Data: Sample Learned Rules**

# 4 Current and Future Research

An under-studied issue in relational data mining is scaling algorithms to very large databases. Most research on ILP and RDM has been conducted in the machine learning and artificial intelligence (AI) communities rather than in the database and systems communities. Consequently, there has been insufficient research on systems issues involved in performing RDM in commercial relational-database systems and scaling algorithms to extremely large datasets that will not fit in main memory. Integrating ideas from systems work in data mining and deductive databases [31] would seem to be critical in addressing these issues.

Related to scaling, we are currently working on efficiently learning complex relational concepts from large amounts of data by using stochastic sampling methods. A major shortcoming of ILP is the computational demand that results from the large hypothesis spaces searched. Intelligently sampling these large spaces can provide excellent performance in much less time [34, 45].

We are also developing algorithms that learn more robust, probabilistic relational concepts represented as stochastic logic programs [25] and variants. This will enrich the expressiveness and robustness of learned concepts. As an alternative to stochastic logic programs, we are working on learning clauses in a constraint logic programming language where the constraints are Bayesian networks [30, 7].

One approach that we plan to further investigate is the use of approximate prior knowledge to induce more accurate, comprehensible relational concepts from fewer training examples [32]. The use of prior knowledge can greatly reduce the burden on users; they can express the "easy" aspects of the task at hand and then collect a small number of training examples to refine and extend this prior knowledge.

Finally, we plan to use active learning to allow our ILP systems to select more effective training examples for interactively learning relational concepts [26]. By intelligently choosing the examples for users to label, better extraction accuracy can be obtained from fewer examples, thereby greatly reducing the burden on the users of our ILP systems.

# 5 Related Work

Although it is the most widely studied, ILP is not the only approach to relational data mining. In particular, other participants in the EELD program are taking alternative RDM approaches to pattern learning for link discovery. This section briefly reviews these other approaches.

## 5.1 Graph-based Relational Learning

Some relational data mining methods are based on learning structural patterns in graphs. In particular, SUBDUE [4, 5] discovers highly repetitive subgraphs in a labeled graph using the minimum description length (MDL) principle. SUBDUE can be used to discover interesting substructures in graphical data as well as to classify and cluster graphs. Discovered patterns do not have to match the data exactly since SUBDUE can employ an inexact graph-matching procedure based on graph edit-distance. SUBDUE has been successfully applied to a number of important RDM problems in molecular biology, geology, and program analysis. It is also currently being applied to discover patterns for link discovery as a part of the EELD project (see `http://ailab.uta.edu/eeld/`). Since relational data for LD is easily represented as labeled graphs, graph-based RDM methods like SUBDUE are a natural approach.

## 5.2 Probabilistic Relational Models

*Probabilistic relational models* (PRM's) [20] are an extension of Bayesian networks for handling relational data. Methods for learning Bayesian networks have also been extended to produce algorithms for inducing PRM's from data [16]. PRM's have the nice property of integrating some of the advantages of both logical and probabilistic approaches to knowledge representation and reasoning. They combine some of the representational expressivity of first-order logic with the uncertain reasoning abilities of Bayesian networks. PRM's have been applied to a number of interesting problems in molecular biology, web-page classification, and analysis of movie data. They are also currently being applied to pattern learning for link discovery as a part of the EELD project.

### 5.3 Relational Feature Construction

One approach to learning from relational data is to first "flatten" or "propositionalize" the data by constructing features that capture some of the relational information and then applying a standard learning algorithm to the resulting feature vectors [21]. PROXIMITY [28] is a system that constructs features for categorizing entities based on the categories and other properties of other entities to which it is related. It then uses an interactive classification procedure to dynamically update inferences about objects based on earlier inferences about related objects. PROXIMITY has been successfully applied to company and movie data. It is also currently being applied to pattern learning for link discovery as a part of the EELD project.

## 6 Conclusions

Link discovery is an important problem in automatically detecting potential threatening activity from large, heterogeneous data sources. The DARPA EELD program is a U.S. government research project exploring link discovery as an important problem in the development of new counter-terrorism technology. Learning new link-discovery patterns that indicate potentially threatening activity is a difficult data mining problem. It requires discovering novel relational patterns in large amounts of complex relational data. Most existing data-mining methods assume flat data from a single relational table and are not appropriate for link discovery. Relational data mining techniques, such as inductive logic programming, are needed. Many other problems in molecular biology [36], natural-language understanding [46], web page classification [9], information extraction [3, 15], and other areas also require mining multi-relational data. However, relational data mining requires exploring a much larger space of possible patterns and performing complex inference and pattern matching. Consequently, current RDM methods are not scalable to very large databases. Consequently, we believe that relational data mining is one of the major research topics in the development of the next generation of data mining systems, particularly those in the area of counter-terrorism.

## Acknowledgments

This work was supported by DARPA EELD Grant F30602-01-2-0571, Vítor Santos Costa and Inês de Castro Dutra are on leave from COPPE/Sistemas, Federal University of Rio de Janeiro and were partially supported by CNPq. We would like to thank the Biomedical Group support staff and the Condor Team at the Computer Sciences Department for their invaluable help with Condor. We also would like to thank Ashwin Srinivasan for his help with the Aleph system.

## References

[1] L. Breiman. Bagging Predictors. *Machine Learning*, 24(2):123–140, 1996.

[2] L. Breiman. Stacked Regressions. *Machine Learning*, 24(1):49–64, 1996.

[3] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 328–334, Orlando, FL, July 1999.

[4] D. J. Cook and L. B. Holder. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255, 1994.

[5] D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[6] W. Cook and G. O'Hayon. Chronology of Russian killings. *Transnational Organized Crime*, 4(2), 2000.

[7] V. S. Costa, D. Page, and J. Cussens. CLP(BN): Constraint logic programming with Bayesian network constraints. Unpublished Technical Note, 2002.

[8] J. Cowie and W. Lehnert. Information extraction. *Communications of the Association for Computing Machinery*, 39(1):80–91, 1996.

[9] M. Craven, D. DiPasquo, D. Freitag, A. K. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1-2):69–113, 2000.

[10] T. G. Dietterich. Machine-learning research: Four current directions. *AI Magazine*, 18(4):97–136, 1998.

[11] I. C. Dutra, D. Page, V. S. Costa, and J. Shavlik. An empirical evaluation of bagging in inductive logic programming. In *Proceedings of the 12th International Conference on Inductive Logic Programming*. Springer-Verlag, September 2002.

[12] S. Džeroski. Relational data mining applications: An overview. In S. Džeroski and N. Lavrač, editors, *Relational Data Mining*. Springer Verlag, Berlin, 2001.

[13] S. Džeroski and N. Lavrač. An introduction to inductive logic programming. In S. Džeroski and N. Lavrač, editors, *Relational Data Mining*. Springer Verlag, Berlin, 2001.

[14] S. Džeroski and N. Lavrač, editors. *Relational Data Mining*. Springer Verlag, Berlin, 2001.

[15] D. Freitag. Information extraction from HTML: Application of a general learning approach. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 517–523, Madison, WI, July 1998. AAAI Press / The MIT Press.

[16] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, 1999.

[17] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kauffmann Publishers, San Francisco, 2001.

[18] D. J. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, Cambridge, MA, 2001.

[19] D. Jensen and H. Goldberg, editors. *AAAI Fall Symposium on Artificial Intelligence for Link Analysis*, Menlo Park, CA, 1998. AAAI Press.

[20] D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 580–587, Madison, WI, July 1998. AAAI Press / The MIT Press.

[21] S. Kramer, N. Lavrač, and P. Flach. Propositionalization approaches to relational data mining. In S. Džeroski and N. Lavrač, editors, *Relational Data Mining*. Springer Verlag, Berlin, 2001.

[22] N. Lavrac and S. Dzeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, 1994.

[23] W. Lehnert and B. Sundheim. A performance evaluation of text-analysis technologies. *AI Magazine*, 12(3):81–94, 1991.

[24] S. J. McKay, P. N. Woessner, and T. J. Roule. Evidence extraction and link discovery (EELD) seedling project, database schema description, version 1.0. Technical Report 2862, Veridian Systems Division, August 2001.

[25] S. Muggleton. Stochastic logic programs. *Journal of Logic Programming*, 2002. To appear.

[26] S. Muggleton, C. Bryant, C. Page, and M. Sternberg. Combining active learning with inductive logic programming to close the loop in machine learning. In S. Colton, editor, *Proceedings of the AISB'99 Symposium on AI and Scientific Creativity (informal proceedings)*, 1999.

[27] S. H. Muggleton, editor. *Inductive Logic Programming*. Academic Press, New York, NY, 1992.

[28] J. Neville and D. Jensen. Iterative classification in relational data. In *Papers from the AAAI-00 Workshop on Learning Statistical Models from Relational Data*, Austin, TX, 2000. AAAI Press / The MIT Press.

[29] NIST. ACE - Automatic Content Extraction. http://www.nist.gov/speech/tests/ace/.

[30] D. Page. ILP: Just do it! In J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. Pereira, Y. Sagiv, and P. Stuckey, editors, *Proceedings of Computational Logic 2000*, pages 25–40. Springer Verlag, 2000.

[31] K. Ramamohanarao and J. Harland. An introduction to deductive database languages and systems. *VLDB Journal*, 3:2, 1994.

[32] B. L. Richards and R. J. Mooney. Automated refinement of first-order Horn-clause domain theories. *Machine Learning*, 19(2):95–131, 1995.

[33] M. K. Sparrow. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks*, 13:251–274, 1991.

[34] A. Srinivasan. A study of two sampling methods for analysing large datasets with ILP. *Data Mining and Knowledge Discovery*, 3(1):95–123, 1999.

[35] A. Srinivasan. *The Aleph Manual*, 2001. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn /Aleph/aleph_toc.html.

[36] A. Srinivasan, S. H. Muggleton, M. J. Sternberg, and R. D. King. Theories for mutagenicity: A study in first-order and feature-based induction. *Artificial Intelligence*, 85:277–300, 1996.

[37] S. Wasserman and K. Faust. *Social Network Analysis: Methods & Applications*. Cambridge University Press, Cambridge, UK, 1994.

[38] P. Williams. Patterns, indicators, and warnings in link analysis: The contract killings dataset. Technical Report 2878, Veridian Systems Division, January 2002.

[39] P. Williams and P. N. Woessner. Nuclear material trafficking: An interim assessment. *Transnational Organized Crime*, 1(2):206–238, 1995.

[40] P. Williams and P. N. Woessner. Nuclear material trafficking: An interim assessment, ridgway viewpoints. Technical Report 3, Ridgway Center, University of Pittsburgh, February 1995.

[41] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, 1999.

[42] P. N. Woessner. Chronology of nuclear smuggling incidents: July 1991-may 1995. *Transnational Organized Crime*, 1(2):288–329, 1995.

[43] P. N. Woessner. Chronology of radioactive and nuclear materials smuggling incidents: July 1991-june 1997. *Transnational Organized Crime*, 3(1):114–209, 1997.

[44] S. Wrobel. Inductive logic programming for knowledge discovery in databases. In S. Džeroski and N. Lavrač, editors, *Relational Data Mining*. Springer Verlag, Berlin, 2001.

[45] F. Zelezny, A. Srinivasan, and D. Page. Lattice-search runtime distributions may be heavy-tailed. In *The Twelfth International Conference on Inductive Logic Programming*. Springer Verlag, July 2002.

[46] J. M. Zelle and R. J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1050–1055, Portland, OR, Aug. 1996.